

Acknowledgement

We are pleased to present the “**Disease prediction**” project and take this opportunity to express our profound gratitude to all those who helped us complete this project.

We thank our college for providing excellent facilities that helped us complete and present this project. We would also like to thank the staff members and lab assistants for permitting us to use computers in the lab as and when required.

We express our deepest gratitude towards our project guide for his/her valuable and timely advice during the various phases of our project. We would also like to thank him/her for providing us with all the proper facilities and support as the project co-coordinator. We would like to thank him/her for support, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules.

We would like to thank all our friends for their smiles and friendship making college life enjoyable and memorable and family members who always stood beside us and provided the utmost important moral support. Finally, we would like to thank everyone who has helped us directly or indirectly in our project.

TABLE OF CONTENT

| Sr. No. | Contents |
|----------|--|
| 1 | <u>PROJECT OVERVIEW</u> |
| 1.1 | Introduction |
| 1.2 | Scope and Objective |
| 1.3 | Modules and their Description |
| 1.4 | Existing System & Proposed System |
| 2 | <u>PROJECT ANALYSIS</u> |
| 2.1 | Gantt Chart |
| 3 | <u>PROJECT LIFECYCLE</u> |
| 3.1 | Project Lifecycle Details |
| 4 | <u>PROJECT DESIGN</u> |
| 4.1 | E-R Diagram |
| 4.2 | Use Case Diagram |
| 4.3 | Sequence Diagram |
| 4.4 | Activity Diagram |
| 4.5 | Data Flow Diagram |
| 4.6 | System Architecture |
| 5 | <u>SNAPSHOTS</u> |
| 5.1 | Project Snapshots |
| 6 | <u>PROJECT IMPLEMENTATION</u> |
| 6.1 | Project Implementation Technology |
| 6.2 | Feasibility Report |
| 7 | <u>CODING</u> |
| 7.1 | Project Coding |
| 8 | <u>TESTING</u> |
| 8.1 | Testing |
| 8.2 | Levels of Testing |
| 8.3 | Test Cases |
| 9 | <u>ADVANTAGES & LIMITATIONS</u> |
| 9.1 | Advantages |
| 9.2 | Limitations |
| 9.3 | Features |

| | |
|-----------|----------------------------|
| 10 | <u>CONCLUSION</u> |
| 10.1 | Project Conclusion |
| 11 | <u>BIBLIOGRAPHY</u> |
| 11.1 | Website Links |

PROJECT OVERVIEW

Introduction

The healthcare sector has emerged as a significant focal point globally, with the prevalence of health problems on the rise. In India, the management of rural healthcare poses a considerable challenge. Effective early diagnosis and treatment of diseases can greatly impact outcomes in these areas. However, physical consultations are often expensive and time-consuming, especially in many regions.

Based on the situation described, we have created an AI Health Bot System using Python. This chatbot is programmed to communicate with users using natural language. The primary objective of this system is to detect diseases by analyzing user-provided symptoms and offering preliminary information about the identified disease before consulting a doctor. To accomplish disease prediction, we have employed the Logistic Regression algorithm.

Scope and Objective

The scope and objective of an AI health bot encompass a broad range of possibilities in the field of healthcare. AI health bots are designed to leverage artificial intelligence technologies to provide various healthcare-related services and support. These bots can be deployed in different settings, such as online platforms, mobile applications, or integrated into existing healthcare systems. Their primary objective is to enhance healthcare accessibility, improve patient outcomes, and streamline healthcare processes.

The AI health bot utilizes natural language processing and machine learning algorithms to engage in conversations with users, collecting information about their symptoms and offering initial assessments. This functionality allows individuals to receive preliminary guidance on the severity of their symptoms and whether immediate medical attention is necessary or if self-care measures would suffice. By efficiently triaging patients and directing them to appropriate care settings, the bot helps alleviate the burden on healthcare providers. Moreover, the bot can detect nearby doctors who specialize in treating the specific disease identified during the conversation, enabling users to conveniently book appointments with healthcare professionals through the same platform.

Modules and their Description

The system comprises 1 major module with their sub-modules as follows:

❖ **User:**

- Registration:
 - The user can register using their basic details.
- Login:
 - They can log in using their credentials.
- Dashboard:
 1. Users can search articles for a specific disease and based on that, articles will be shown.
 2. Users can enter the symptoms they are dealing with and the system will predict the disease accordingly.
 3. Based on the disease detected, doctors will be shown concerning the user's location (Area).
 4. The user will be able to book an appointment with the doctor.
- Appointment Details:
 1. View Status: The user will be able to view the status of the appointment as accepted or rejected.
 2. View treatment details:
 - Prescription:
The user will be able to view the prescription.
 - Test Results:
The user will need to upload the test results if required.
 - Video Call:
The user can request the doctor for a video call.
- Edit profile:
 - The user will be able to edit their profile.
- Give feedback/Ask queries:

- They can even give feedback or ask any query they want.

❖ **Admin:**

- Login:
 - The admin can log in using their credentials.
- View Users:
 - They can view the user details.
- View Doctors:
 - They can manage doctor requests and view their details.
- Manage Data:
 - The admin will add information regarding the disease that is
 - Add articles related to the disease
 - Add the disease name
 - Add description
 - Add a summary of the article
 - Add article link (this link will be clickable and used from online sources only)
- View Feedback/answer query:
 - The admin will be able to view feedback and answer queries asked by the users and doctors.

❖ **Doctor:**

- Registration:
 - The doctor will need to register first to access the system.
- Login:
 - They can log in further if the admin accepts their requests.
- View Users:
 - They can view the user details.
- Appointments Details:
 1. Accept or reject requests:

- The doctor can view the users 'appointments and accept or reject the requests.
- 2. Give treatment:
 - The doctor can add medicines and lab test details for every user as required.
- 3. Medicines:
 - If lab tests are uploaded, then they can further add medicines based on the results.
- 4. Video call:
 - The doctor can consult the patient over a video call if possible.
- Edit Profile:
 - They can edit their profile if they want.
- Give Feedback/Ask Query:
 - They can also provide feedback or ask any query if required.

Existing System & Proposed System

❖ Problem with the current scenario

- Healthcare systems often suffer from fragmentation and lack of coordination among different providers and healthcare settings.
- Lack of interoperability and information sharing among healthcare technologies and systems further exacerbates this issue.

❖ Drawbacks of the existing system

- In the existing systems, there are a limited number of diseases in the dataset.
- Also, technical issues like text messages are not accurate in the existing system.

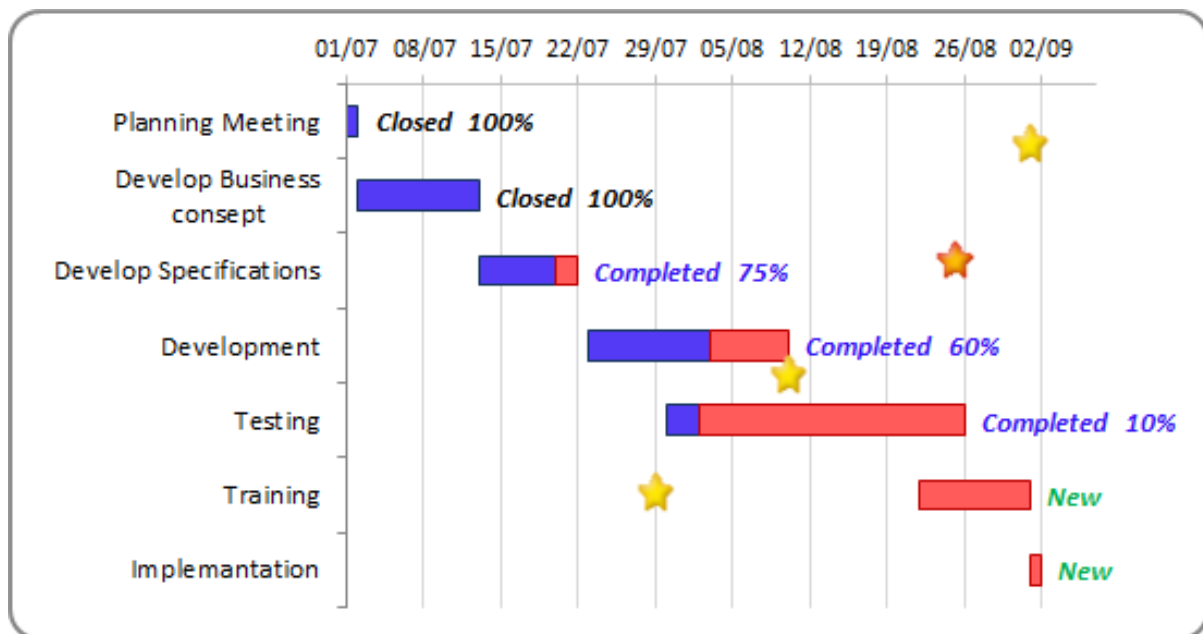
PROPOSED SYSTEM

- The system comprises 3 major modules: User, Admin and Doctor
- The user interface of the system offers several features for users. The registration process allows users to sign up by providing their basic details. Once registered, users can log in using their credentials. The dashboard is the main hub for users, providing various functionalities. Firstly, users can search for articles related to specific diseases, and relevant articles will be displayed. Secondly, users can enter their symptoms, and the system will predict the corresponding disease. Based on the detected disease, doctors available in the user's area will be shown. Users can then proceed to book an appointment with a doctor.
- The appointment details section provides additional information. Users can view the status of their appointments, whether they have been accepted or rejected. They can also access treatment details, including the prescription provided by the doctor. If necessary, users may be required to upload test results. Additionally, users have the option to request a video call with the doctor for a virtual consultation. The system also allows users to edit their profiles and provide feedback or ask any queries they may have.
- The admin interface provides specific functionalities for administrative purposes. The admin can log in using their credentials to access the system. They have the ability to view user details, including their registration information. Admins can also manage doctor requests and view the details of registered doctors.
- In terms of data management, the admin is responsible for adding disease-related information. This includes adding articles pertaining to

diseases, disease names, descriptions, summaries of articles, and clickable links to online sources. Furthermore, the admin can view feedback and answer queries submitted by users and doctors.

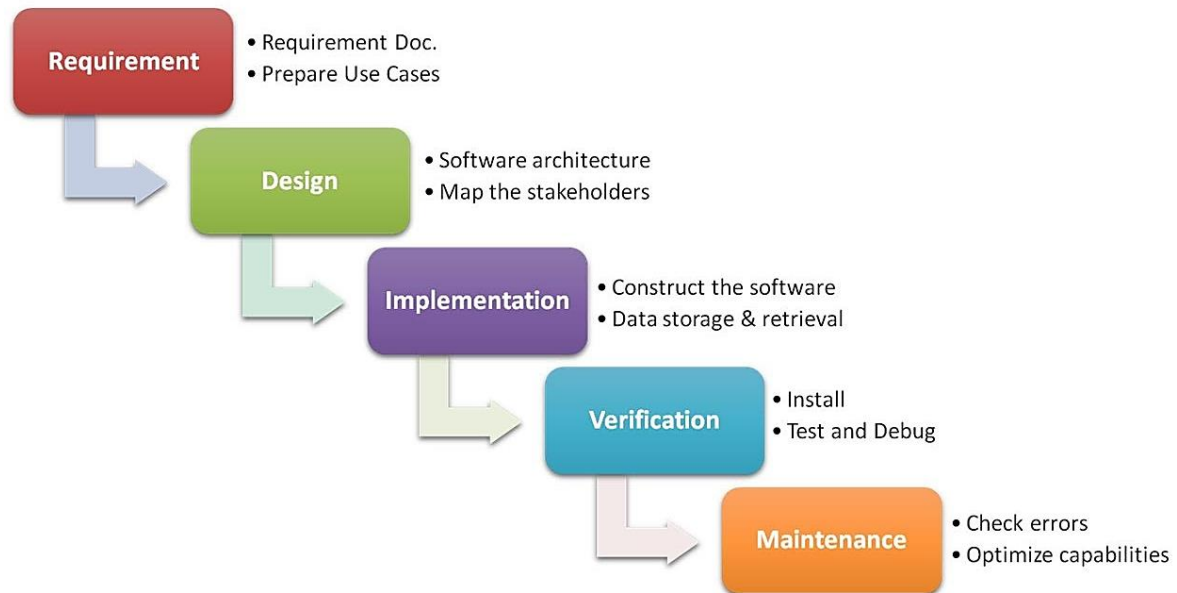
- To access the system, doctors must register and await approval from the admin. Once approved, they can log in using their credentials. Doctors have the ability to view user details, enabling them to access relevant patient information.
- In the appointment details section, doctors can review users' appointment requests and accept or reject them accordingly. They can also provide treatment by adding medications and lab test details for each user as required. If lab test results have been uploaded, doctors can prescribe medicines based on the test outcomes. Additionally, doctors can conduct video calls to consult with patients, if feasible. Doctors also have the option to edit their profiles and provide feedback or ask queries if needed.
- Logistic Regression algorithms have been employed for predicting the likelihood of a user having a particular disease based on their symptoms or other relevant factors. The chatbot can collect information from the user, such as symptoms, medical history, and demographic details. This data can be fed into a logistic regression model trained on historical patient data, enabling the chatbot to estimate the probability of a specific disease.
- In this project, HTML, CSS, and JavaScript are used on the front end, and Python is used on the back end. Here, we have used the Django framework and MySQL Database.

Gantt Chart



Project Lifecycle Details

Waterfall Model

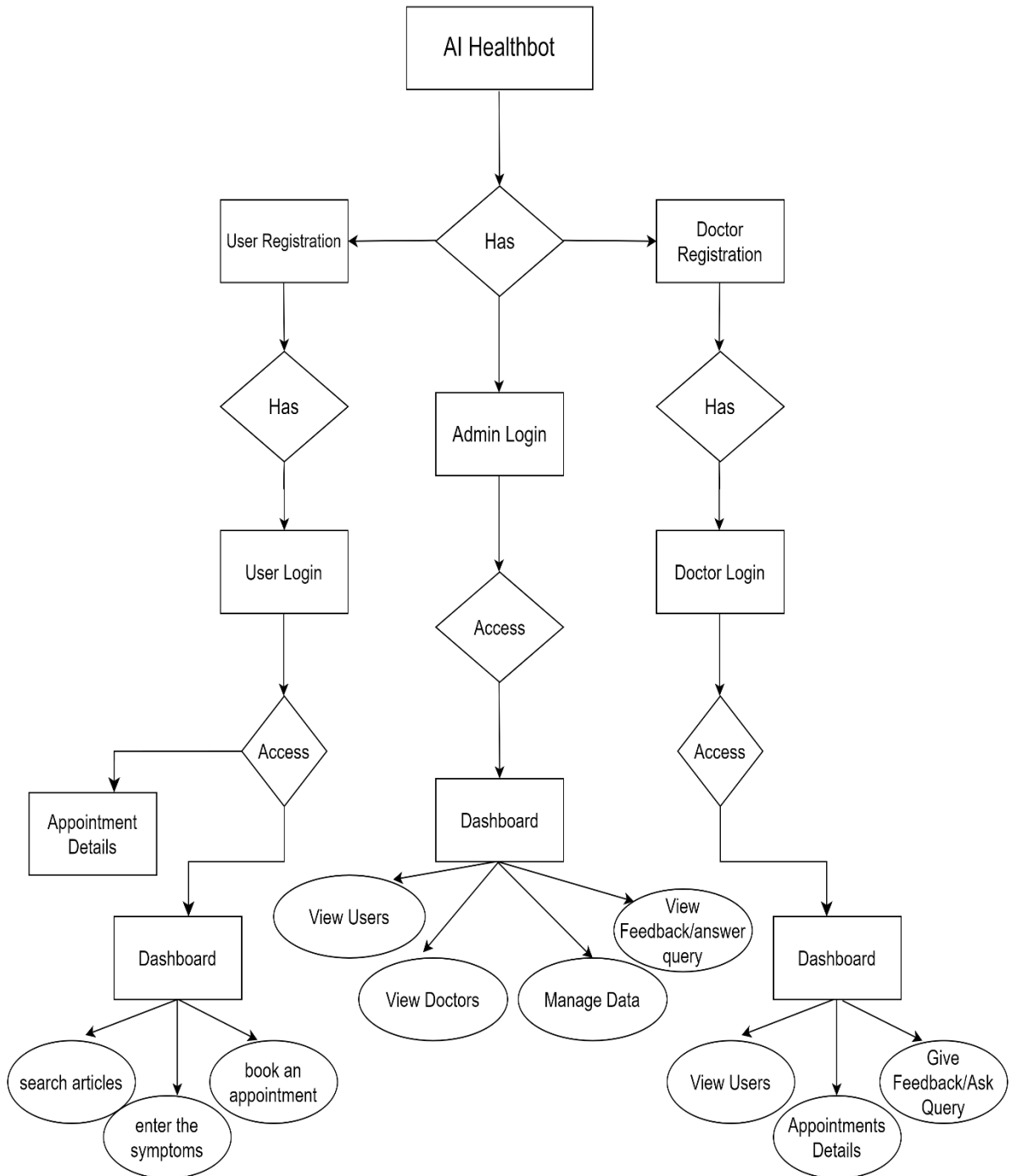


Description

The Waterfall Model is a linear sequential flow in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle requirement changes. The waterfall approach is the earliest approach that was used for software development.

PROJECT DESIGN

E-R Diagram



Use Case Diagram

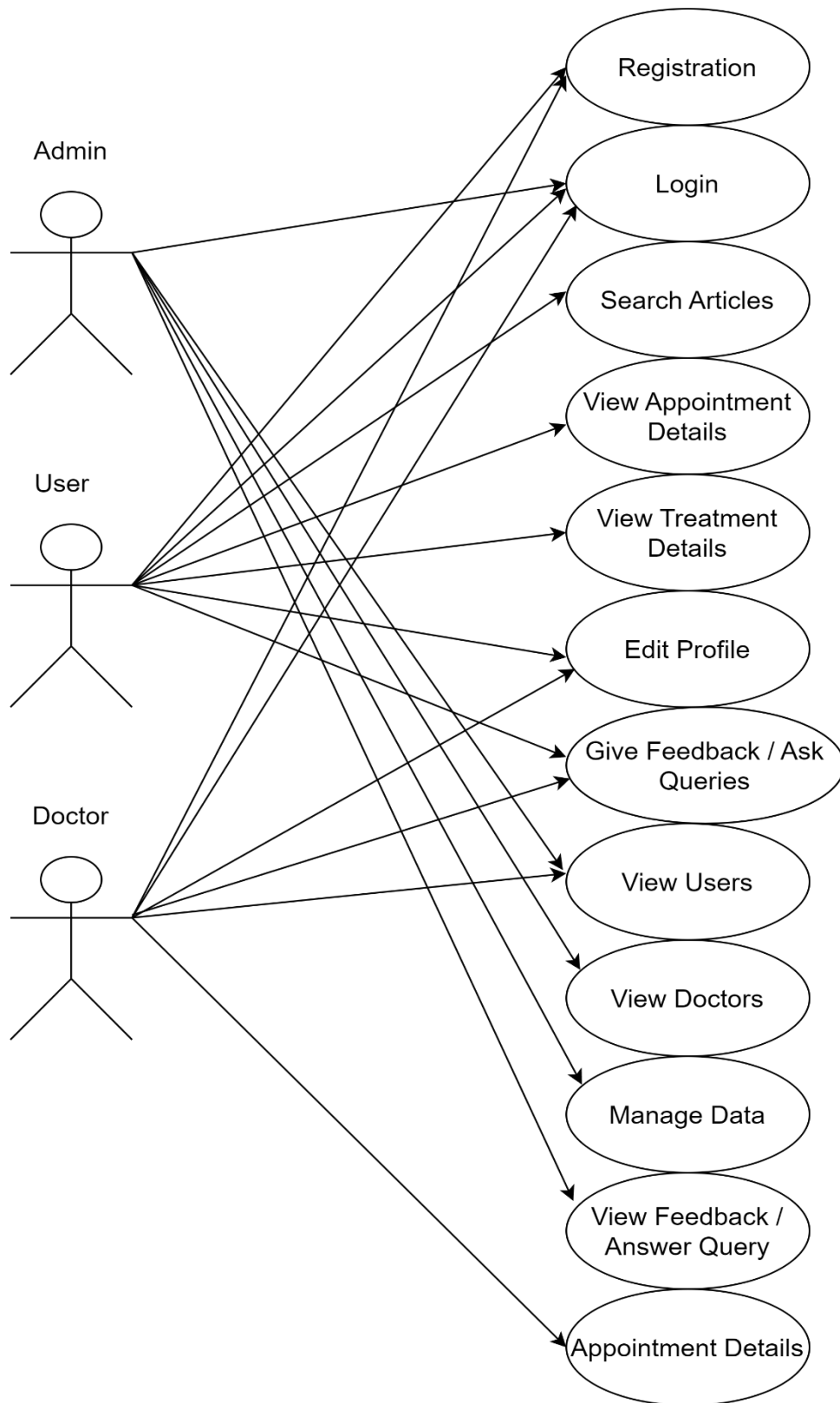


Fig. Use Case Diagram

Sequence Diagram

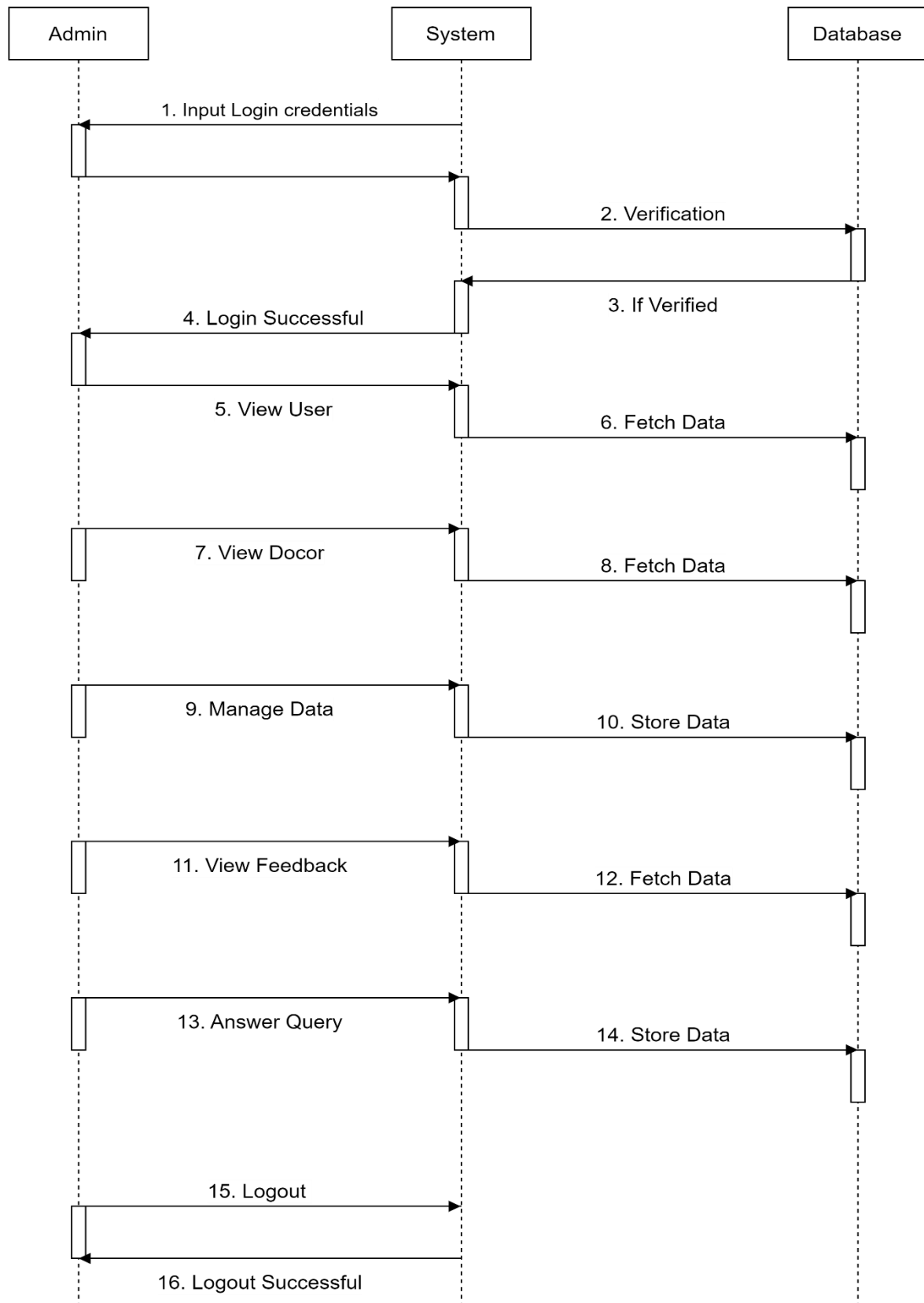


Fig. Sequence Diagram of Admin

Sequence Diagram

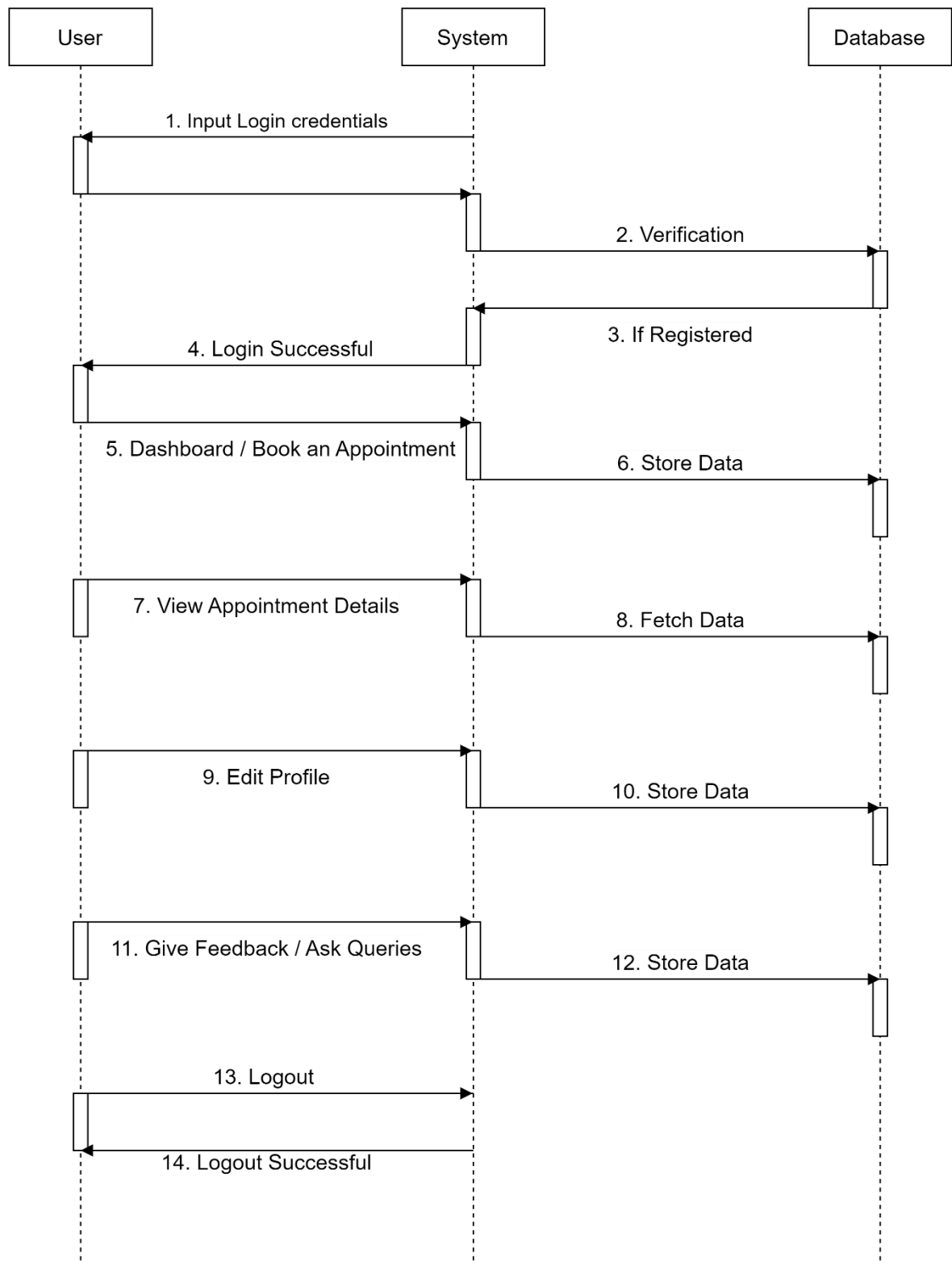


Fig. Sequence Diagram of User

Sequence Diagram

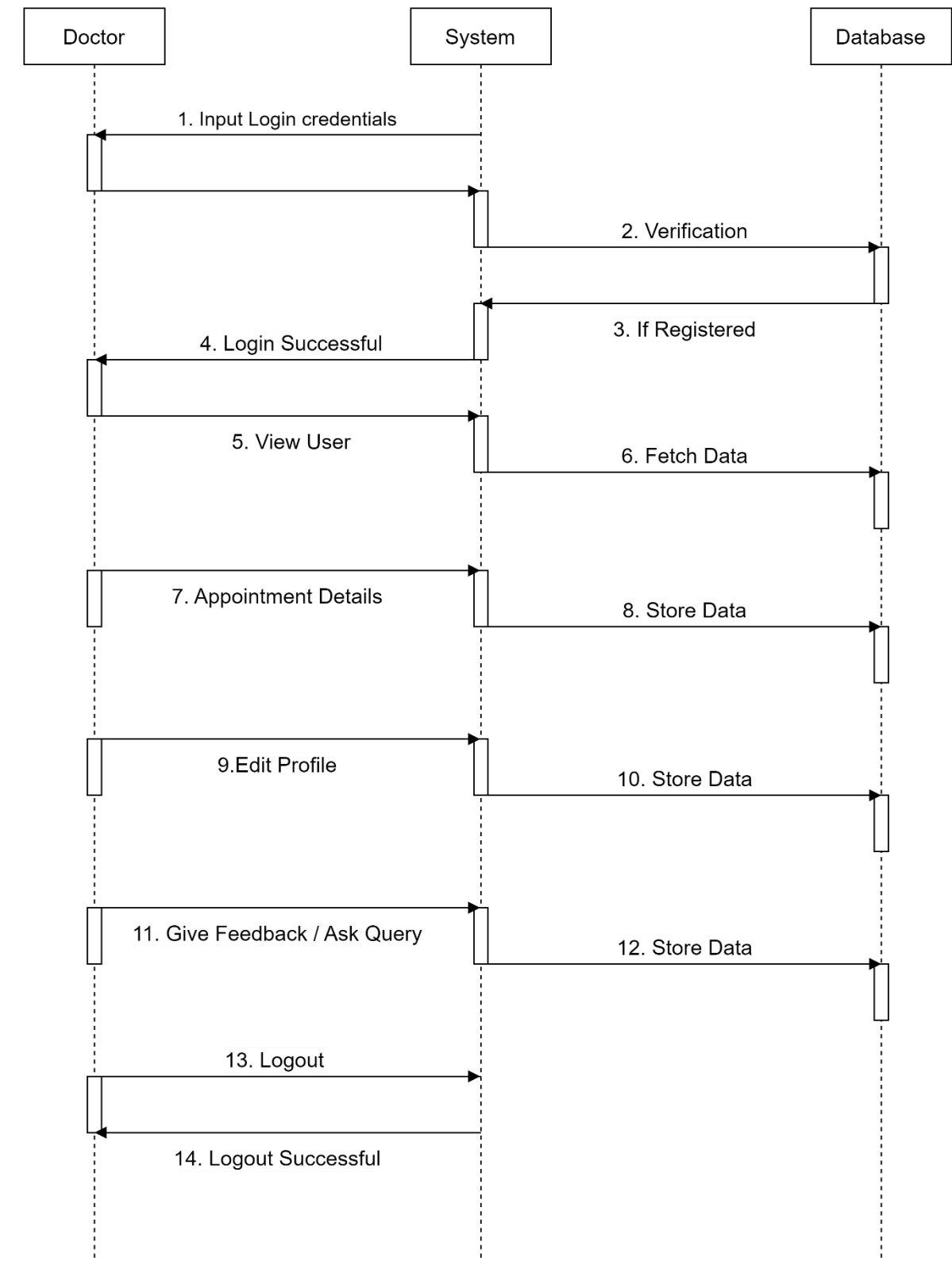


Fig. Sequence Diagram of Doctor

Activity Diagram

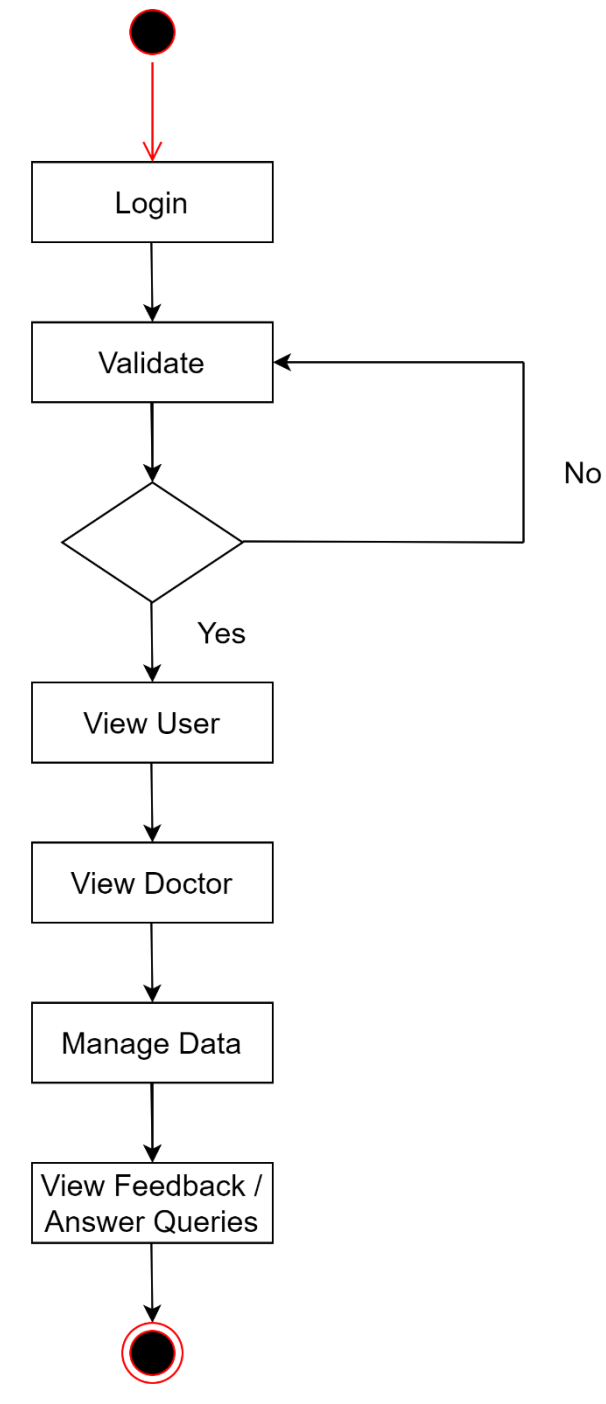


Fig. Activity Diagram of Admin

Activity Diagram

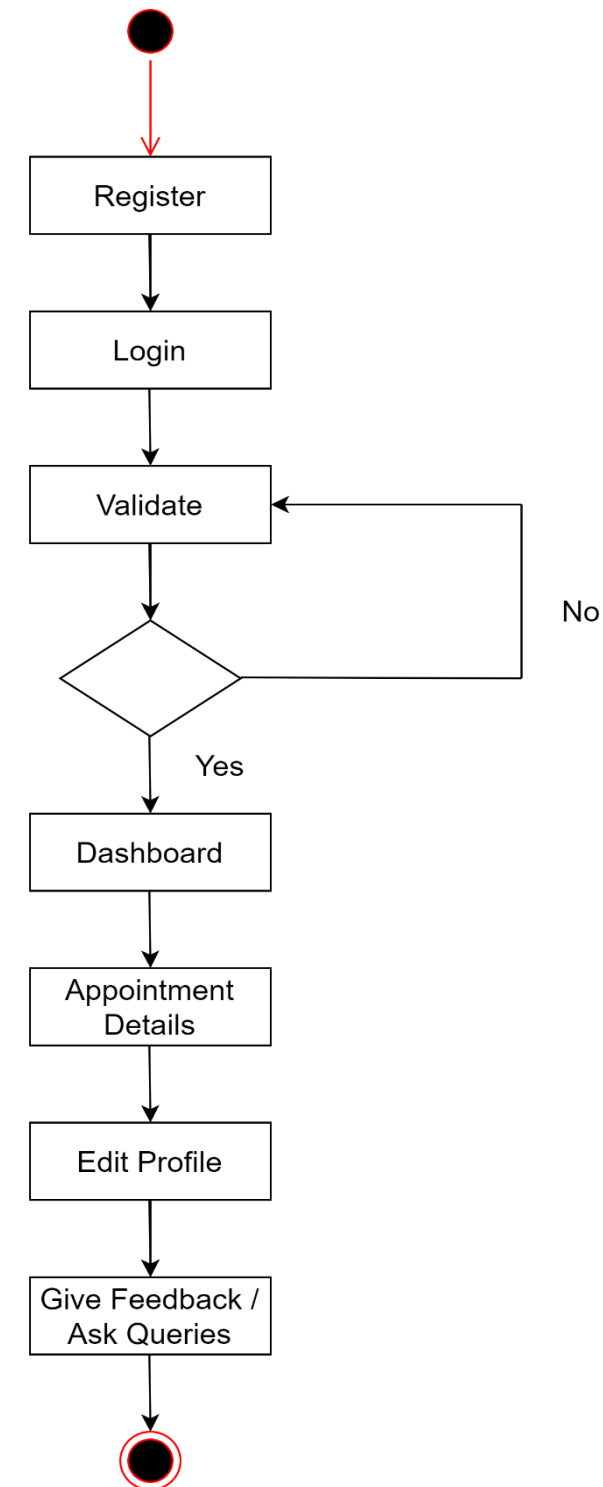


Fig. Activity Diagram of User

Activity Diagram

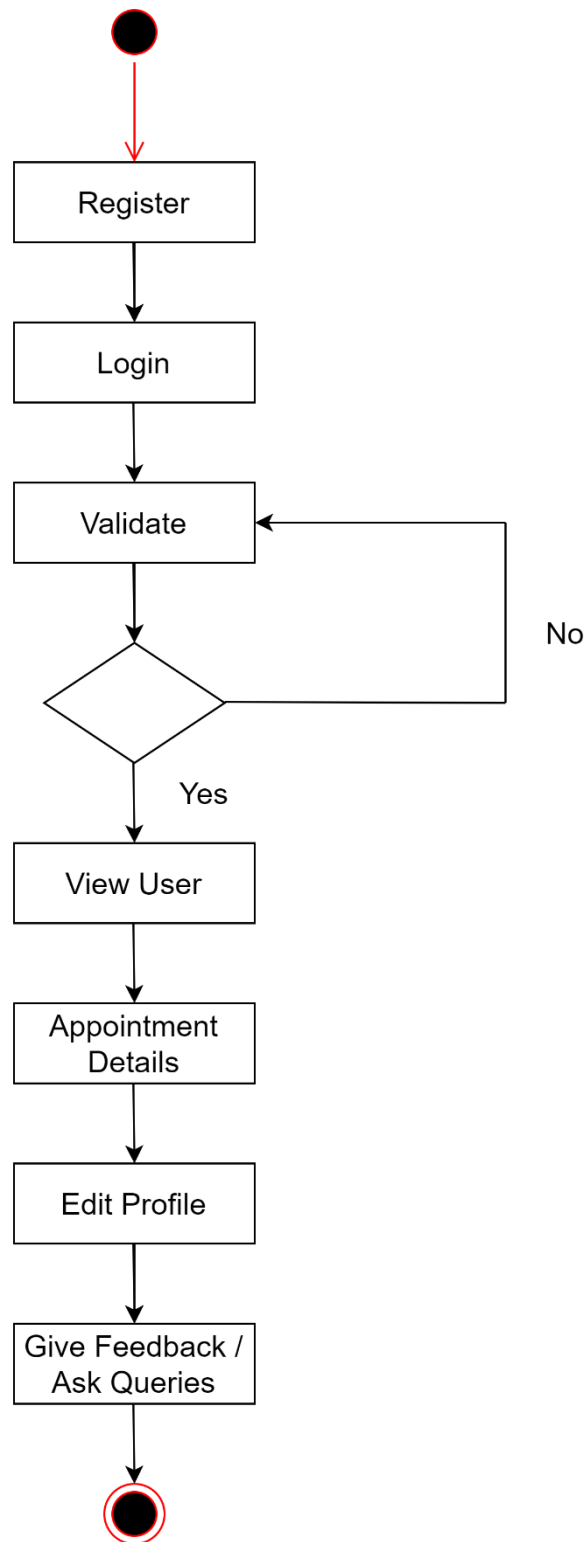
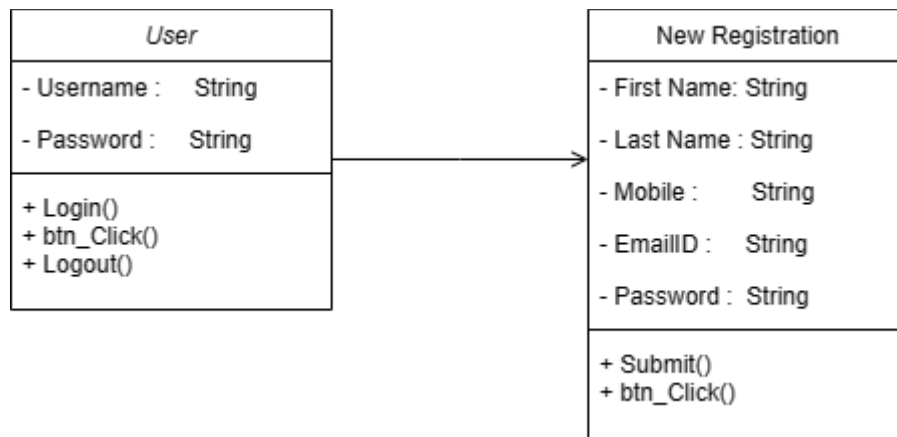


Fig. Activity Diagram of Doctor

Class Diagram



Data Flow Diagrams (DFDs)

A data flow diagram is a graphical tool used to describe and analyze the movement of data through a system. These are the central tool and the basis for the other components' development. The transformation of data from input to output, through processing, may be described logically and independently of physical components associated with the system. These are known as logical data flow diagrams. The physical data flow diagrams show the actual implementation and movement of data between people, departments and workstations. A full description of a system consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labelled with a descriptive name. The process is further identified with a number that will be used for identification purposes. The development of DFDs is done on several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD at the next level. The top-level diagram is often called a context diagram. It consists of a single process bit, which plays a vital role in studying the current system. The process in the context level diagram is exploded into another process at the first level DFD.

The idea behind the explosion of a process into more processes is that understanding at one level of detail is exploded into greater detail at the next level. This is done until the further explosion is necessary and an adequate amount of detail is described for analysts to understand the process.

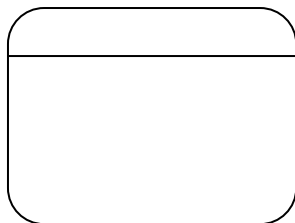
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this led to the modular design.

A DFD also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

In the DFD, there are four symbols

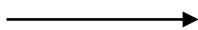
1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



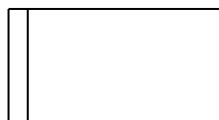
The process that transforms data flow.



Source or Destination of data



Data flow



Data Store

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFDs:

1. Process should be named and numbered for easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from the source to the destination although they may flow back to the source. One way to indicate this is to draw a long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower-level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of the data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out.

Missing interface redundancies and like are then accounted for often through interviews.

SALIENT FEATURES OF DFDs

1. The DFD shows the flow of data, not of control loops and decisions are controlled considerations that do not appear on a DFD.

2. The DFD does not indicate the time factor involved in any process whether the data flows take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL:

In the Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing labels including an identification of the technology used to process the data. Similarly, data flows and data stores are often labelled with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects of the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user was completely happy with the functionality of the current system but had problems with how it was implemented typically the new

logical model will differ from current the logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFDS

PROCESS

- 1) No process can have only outputs.
- 2) No process can have only inputs. If an object has only inputs, then it must be a sink.
- 3) A process has a verb phrase label.

DATASTORE

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into the data store
- 3) A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data.

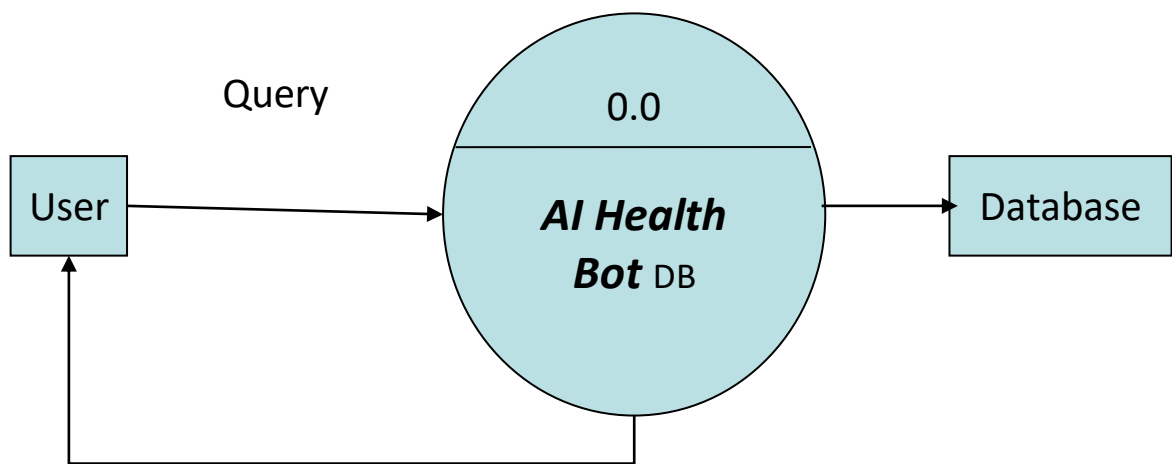
- 1) Data cannot move directly from a source to sink it must be moved by a process

- 2) A source and /or sink has a noun phrase label

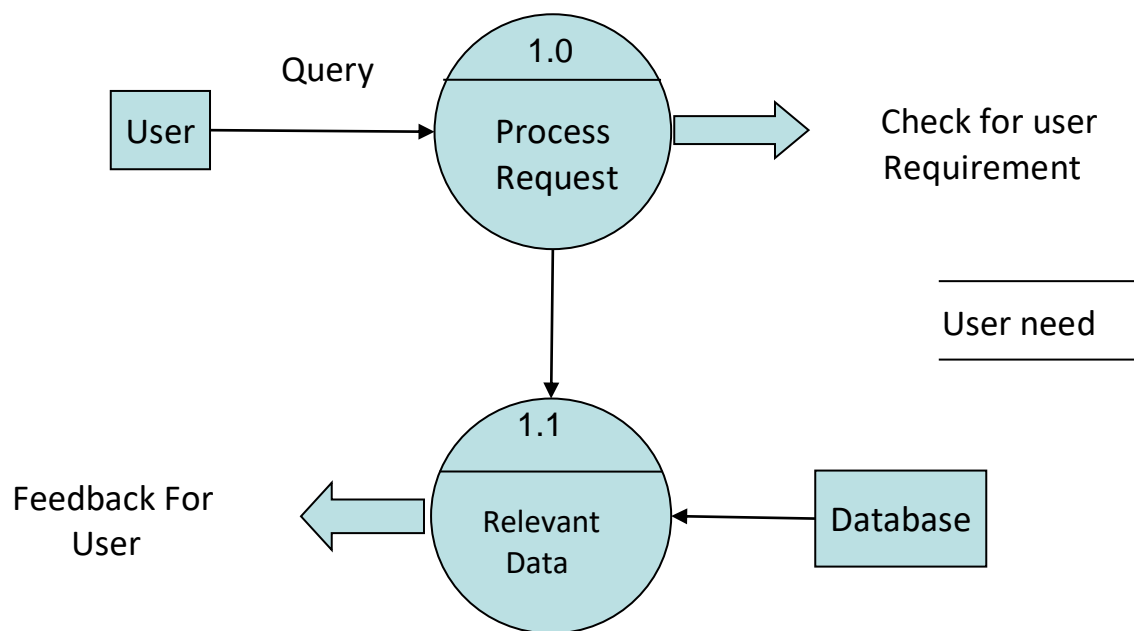
DATA FLOW

- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different times.
- 2) A join in DFD means that the same data comes from any of two or more different processes data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow and produce some other data flow that returns the original data to the beginning process.
- 4) A Data flow to a data store means to update (delete or change).
- 5) A data Flow from a data store means to retrieve or use.

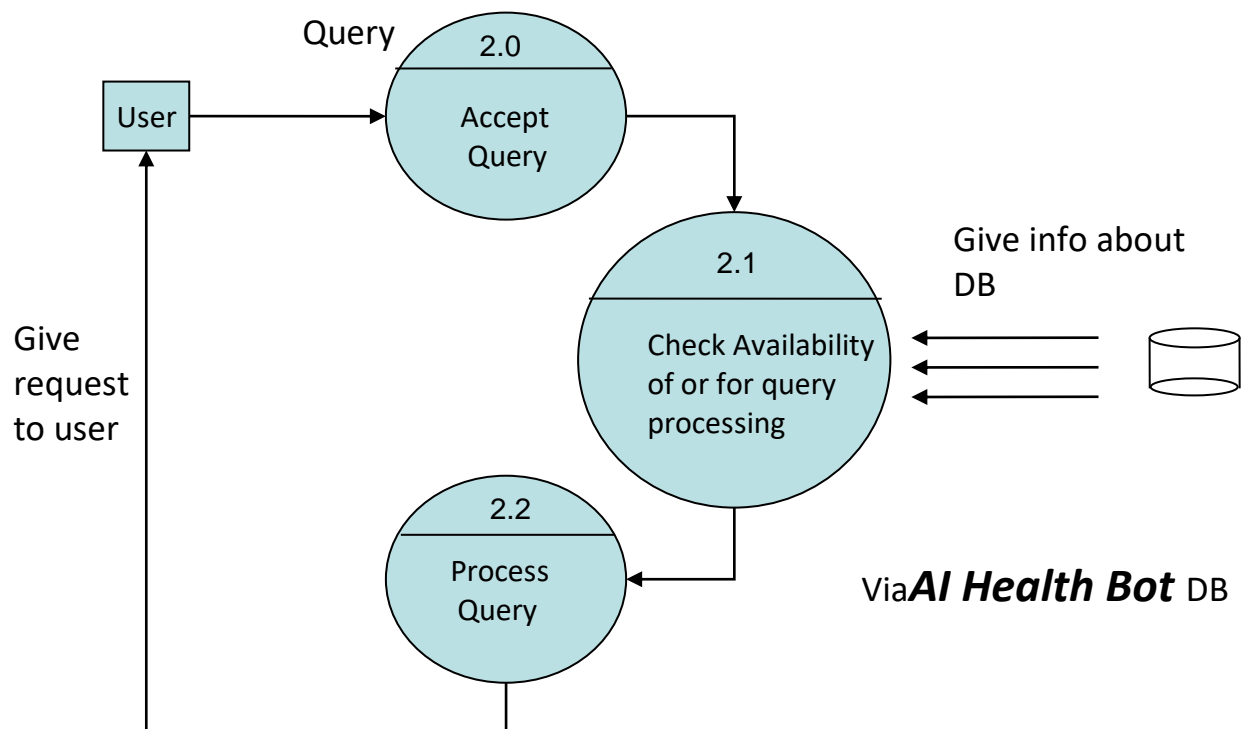
Data Flow Diagrams (DFDs)



DATABASE DETAIL



LEVEL 1 DFD

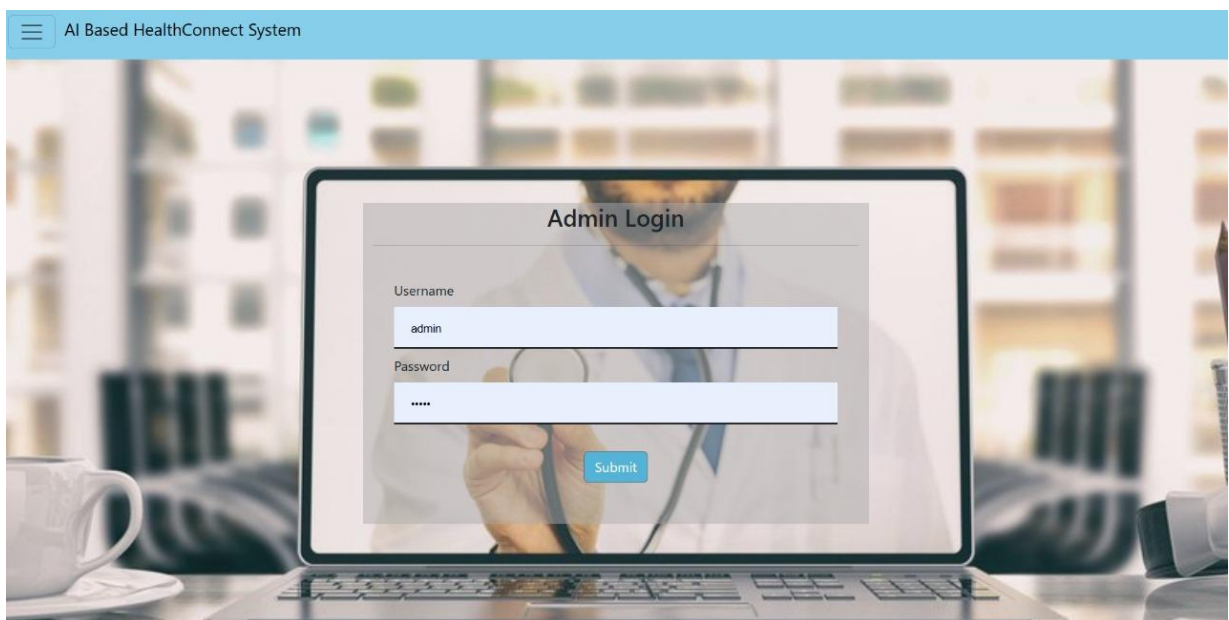
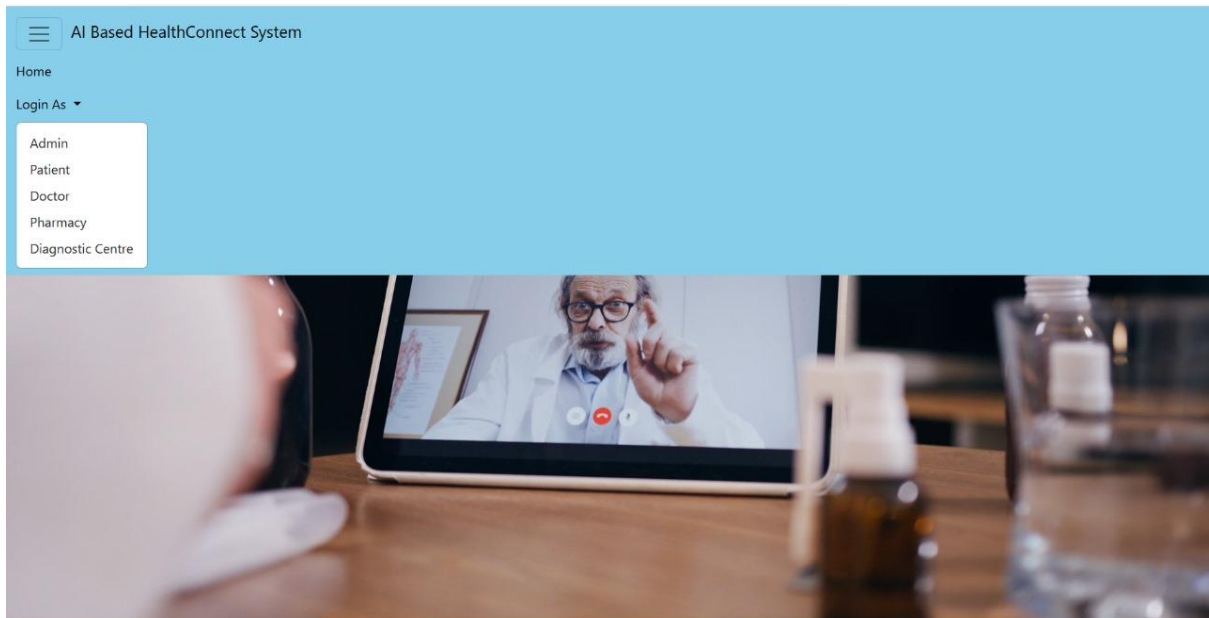


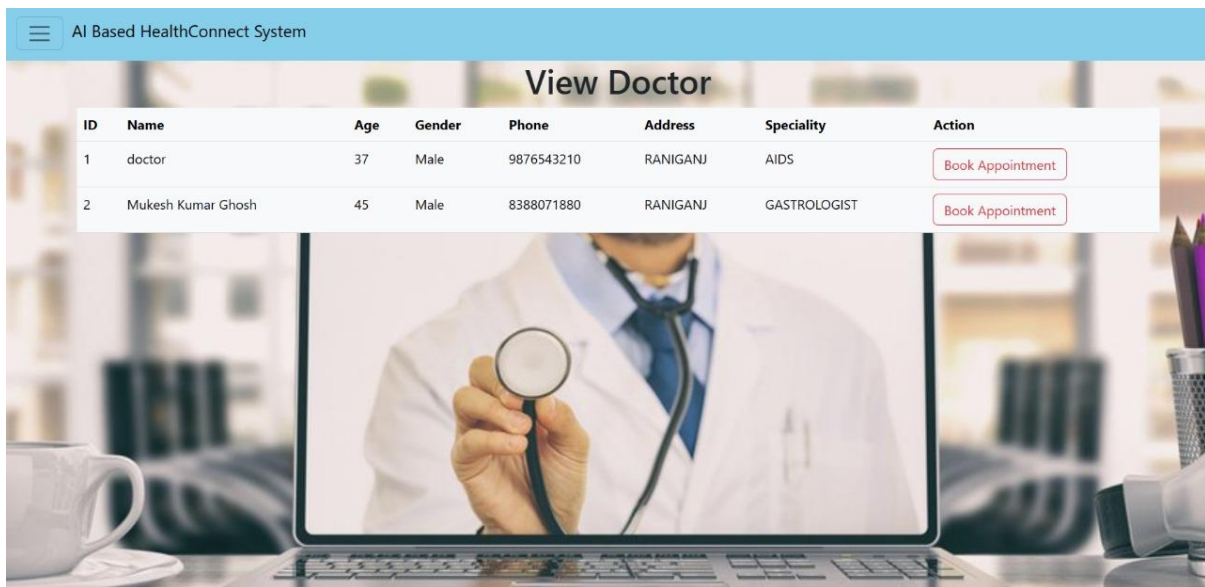
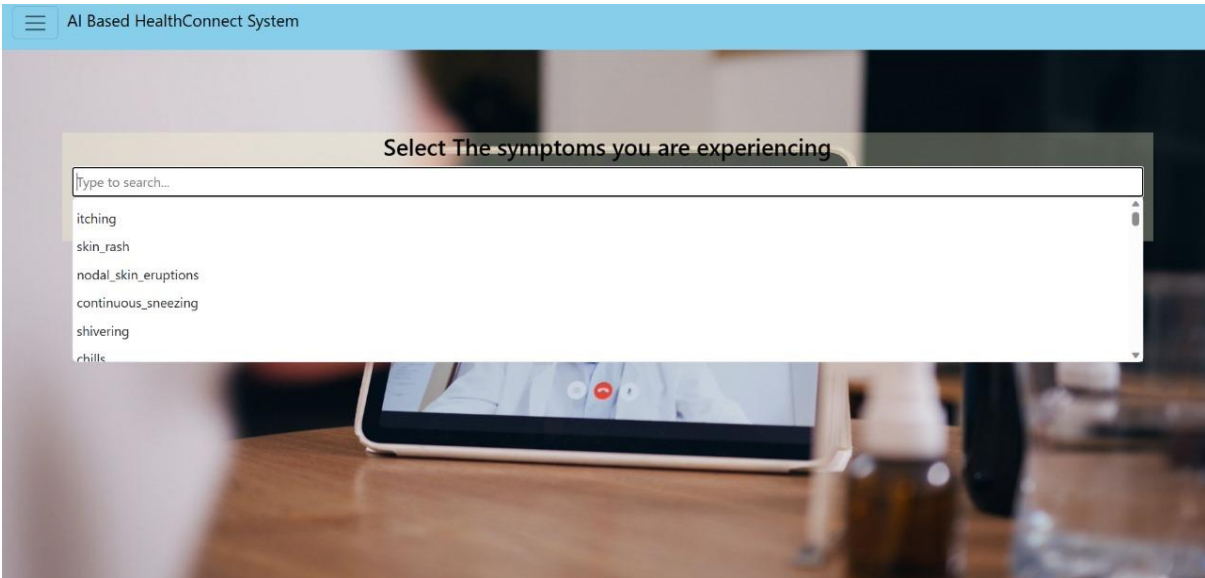
LEVEL 2 DFD: PREDICTION

System Architecture



Snapshots





Medicines



Paracetamol is the most widely used over-the-counter (OTC) medication for the treatment of fever and pain. It provides fast and effective relief from headaches, toothaches, backaches, menstrual pain, musculoskeletal pain associated with/without fever. Paracetamol decreases the intensity of pain signals to the brain and may also block the release of substances called prostaglandins that increases pain and body temperature.

Price: Rs10

Pharmacy: pharmacy

[View Medicine](#)



PROJECT IMPLEMENTATION

Project Implementation Technology

The Project is designed and developed in Django Framework. We used Django Framework for coding the project. Created and maintained all databases in MySQL Server, in that we create tables, and write queries for store data or records of the project.

I. Hardware Requirement

i. Laptop or PC

- Windows 7 or higher
- I3 processor system or higher
- 4 GB RAM or higher
- 100 GB ROM or higher

II. Software Requirement

i. Laptop or PC

- Python
- Sublime Text Editor
- XAMP Server

OVERVIEW OF TECHNOLOGIES USED

INTRODUCTION

Python is a powerful multi-purpose programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. This is a comprehensive guide on how to get started in Python, why you should learn it and how you can learn it. However, if you know other programming languages and want to quickly get started with Python. Python is a general-purpose language. It has a wide range of applications from Web development (like Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and the length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

Features of Python Programming:

- A simple language which is easier to learn
Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like C++, Java, and C#. Python makes programming fun and allows you to focus on the solution rather than syntax.
If you are a newbie, it's a great choice to start your journey with Python.
- Free and open-source
You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software written in it, but you can also even make changes to Python's source code.
Python has a large community constantly improving it in each iteration.
- Portability
You can move Python programs from one platform to another, and run them without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

- **Extensible and Embeddable**
Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.
- **A high-level, interpreted language**
Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.
- **Large standard libraries to solve common tasks**
Python has several standard libraries which makes the life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use the MySQLdb library using `import MySQLdb`. Standard libraries in Python are well-tested and used by hundreds of people. So, you can be sure that it won't break your application.
- **Object-oriented**
Everything in Python is an object. Object-oriented programming (OOP) helps you solve complex problems intuitively. With OOP, you can divide these complex problems into smaller sets by creating objects.

Features of Python Programming:

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management Systems) that are built on Python. Some of the popular platforms for creating Web Apps are Django, Flask, Pyramid, Plone, and Django CMS.

Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like SciPy and NumPy that are used in general-purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.

Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.

However, Python is a great language for creating prototypes. For example, You can use Pygame (a library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies.

It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

Syntax Overview

Simple Elegant Syntax

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

```
a = 2  
  
b = 3  
  
sum = a + b  
  
print(sum)
```

Even if you have never programmed before, you can easily guess that this program adds two numbers and prints them.

Not overly strict

You don't need to define the type of a variable in Python. Also, it's not necessary to add a semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

The expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of the Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed at how much you can do with Python once you learn the basics.

Great Community and Support

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck. Some of them are:

Learn Python subreddit

Google Forum for Python

Python Questions - Stack Overflow

Django documentation

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Features of Django

- Rapid Development
- Secure
- Scalable
- Fully loaded
- Versatile
- Open Source
- Vast and Supported Community

Rapid Development

Django was designed to make a framework which takes less time to build web applications. The project implementation phase is very time taken but Django creates it rapidly.

Secure

Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request

forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable

Django is scalable in nature and can quickly and flexibly switch from small to large-scale application projects.

Fully loaded

Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Versatile

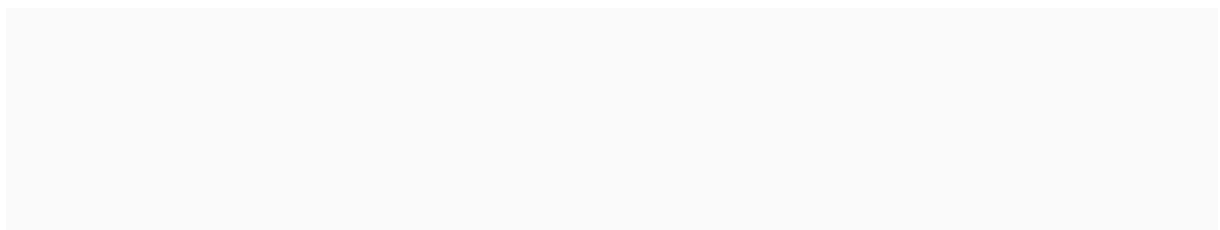
Django is versatile in nature which allows it to build applications for different-different domains. Nowadays, Companies are using Django to build various types of applications like content management systems, social networks sites or scientific computing platforms etc.

Open Source

Django is an open-source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

Vast and Supported Community

Django is one of the most popular web frameworks. It has a widely supportive community and channels to share and connect.



WAMP Server

Introduction

WAMP is a Windows OS-based program that installs and configures Apache web server, MySQL database server, PHP scripting language, phpMyAdmin (to manage MySQL databases), and SQLiteManager (to manage SQLite databases). WAMP is designed to offer an easy way to install Apache, PHP and MySQL packages with an easy-to-use installation program instead of having to install and configure everything yourself. WAMP is so easy because once it is installed it is ready to go. You don't have to do any additional configuring or tweaking of any configuration files to get it running.

There are usually two reasons why someone chooses to install WAMP. They are looking to install WAMP for development purposes or to run their server.

WAMP Server Contains

PHP Admin

Allows you to change or add users and for making new databases
phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL. The most frequently used operations are supported by the user interface (managing databases, tables, fields, relations, indexes, users, permissions, etc.), while you still can directly execute any SQL statement.

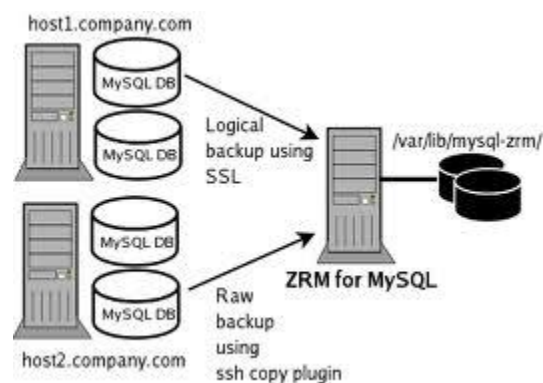
Features

- Intuitive web interface
- Support for most MySQL features:
 - Browse and drop databases, tables, views, fields and indexes.
 - Create, copy, drop, rename and alter databases, tables, fields and indexes.
 - Maintenance server, databases and tables, with proposals on server configuration.
 - Execute, edit and bookmark any SQL statement, even batch queries.
 - Manage MySQL users and privileges
 - Manage stored procedures and triggers.
- Import data from CSV and SQL
- Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LATEX and others
- Administering multiple servers
- Creating PDF graphics of your database layout
- Creating complex queries using Query-by-example (QBE)
- Searching globally in a database or a subset of it
- Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as an image or download-link
- And much more...

SQL Server and Database System



SQL Server is a relational database management system from Microsoft that's designed for the enterprise environment. SQL Server runs on T-SQL (Transact -SQL), a set of programming extensions from Sybase and Microsoft that add several features to standard SQL, including transaction control, exception and error handling, row processing, and declared variables.



Generically, any database management system (DBMS) can respond to queries from client machines formatted in the SQL language. When capitalized, the term generally refers to either of the two database management products from Sybase and Microsoft. Both companies offer client-server DBMS products called SQL Server.

Using WAMP as a Development Server

You can use WAMP to develop and test websites locally on your computer instead of having to get a web hosting account to develop with. Most people will be using WAMP for development purposes such as learning how to create websites with HTML, PHP, and MySQL.

Using WAMP as a Production Server

WARNING: WAMP was designed to be a testing and development server, not an actual production server. WAMP does not come with any real security in place so it offers no protection from any kind of attack. Any 10-year-old with access to the internet can easily hack your WAMP server.

If your website(s) have highly sensitive data (such as credit card numbers, social security numbers, user ids, passwords, etc.), you need to consider this before you put this information online. Unless you are an experienced system administrator and can configure WAMP to be more secure, you should never use WAMP for a production server.

MySQL Configuration

To begin MySQL installation, first download the latest version of Essentials as an MSI package.

During MySQL installation, select Typical installation and use default configuration values except for Sign-Up where you probably want to select Skip Sign-Up. When Setup Wizard is completed, make sure the option Configure the MySQL Server now is set. For MySQL Server Instance Configuration, select Standard Configuration. Next, you must set the option Include Bin Directory in Windows PATH. This setting is crucial, otherwise, a required library, libMySQL.dll, will not be found later during Apache startup.

Finally, enter a proper root password. There is no need to neither enable remote root access nor create an Anonymous Account.

Please inspect messages during MySQL startup and verify that MySQL has been started successfully. Then, you must reboot the system. Otherwise, the required librarylibMySQL.dll will not be found during Apache startup when Apache is trying to load Apache's PHP module and Apache will, perhaps a bit confusingly, complain that it is unable to load the PHP's MySQL library, php_mysql.dll. Therefore, it is necessary to reboot the system at this stage and then continue to the PHP configuration.

PHP Configuration

[PHP](#) for Windows must be installed from the [zip package](#), *not* using the installer because the installer does not work correctly when setting up the configuration files. Download the latest Windows binary version from the 5.x release series.

Create folder C:\Program Files\PHP5 and unzip the downloaded package there. Then, in folder C:\Program Files\PHP5 you need to copy the file php.ini-recommended as php.ini and make two changes to the php.ini file. Change extension_dir to:

```
extension_dir = "C:/Program Files/PHP5/ext/"
```

and also uncomment the following line:

```
extension=php_mysql.dll
```

That is all that is needed for PHP configuration. Additionally, however, if you wish to run PHP from the command line it would be useful to add its installation directory to Windows PATH but for WAMP to operate it is not required. After Apache is installed and configured, also PHP configuration can be tested.

Coding

Codes are provided in the repository

FEASIBILITY REPORT

A Feasibility Study is a high-level capsule version of the entire process intended to answer several questions: What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? A feasibility study is conducted once the problem is clearly understood. A feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system.

The following feasibilities are considered for the project to ensure that the project is viable and it does not have any major obstructions. Feasibility study encompasses the following things:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

In this phase, we study the feasibility of all proposed systems and pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows.

Technical Feasibility

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not.

Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained. The system can be feasible because of the following grounds:

- All necessary technology exists to develop the system.
- This system is too flexible and it can be expanded further.
- This system can give guarantees of accuracy, ease of use, reliability and data security.
- This system can give instant responses to inquiries.

Our project is technically feasible because all the technology needed for our project is readily available.

Operating System : Windows 7 or higher

Languages : python

Database System : My SQL 5.6

Documentation Tool : MS - Word

❖ Economic Feasibility

Economically, this project is completely feasible because it requires no extra financial investment and concerning time, it's completely possible to complete this project in 6 months.

In this step, we verify which proposal is more economical. We compare the financial benefits of the new system with the investment. The new system is economically feasible only when the financial benefits are more than the investments and expenditures. Economic Feasibility determines whether the project goal can be within the resource limits allocated to it or not. It must determine whether it is worthwhile to process the entire project or whether the benefits obtained from the new system are not worth the costs. Financial benefits must be equal to or exceed the costs. In this issue, we should consider:

- The cost to conduct a full system investigation.
- The cost of h/w and s/w for the class of application being considered.
- The development tools.
- The cost of maintenance etc...

Our project is economically feasible because the cost of development is very minimal when compared to the financial benefits of the application.

❖ Operational Feasibility

In this step, we verify different operational factors of the proposed systems like manpower, time etc., and whichever solution uses less operational resources, is the best operationally feasible solution. The solution should also be operationally possible to implement. Operational Feasibility determines if the proposed system satisfied user objectives and could be fitted into the current system operation.

- The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements.
- The clients have been involved in the planning and development of the system.
- The proposed system will not cause any problems under any circumstances.

Our project is operationally feasible because the time requirements and personnel requirements are satisfied. We are a team of four members and we worked on this project for three working months.

TESTING

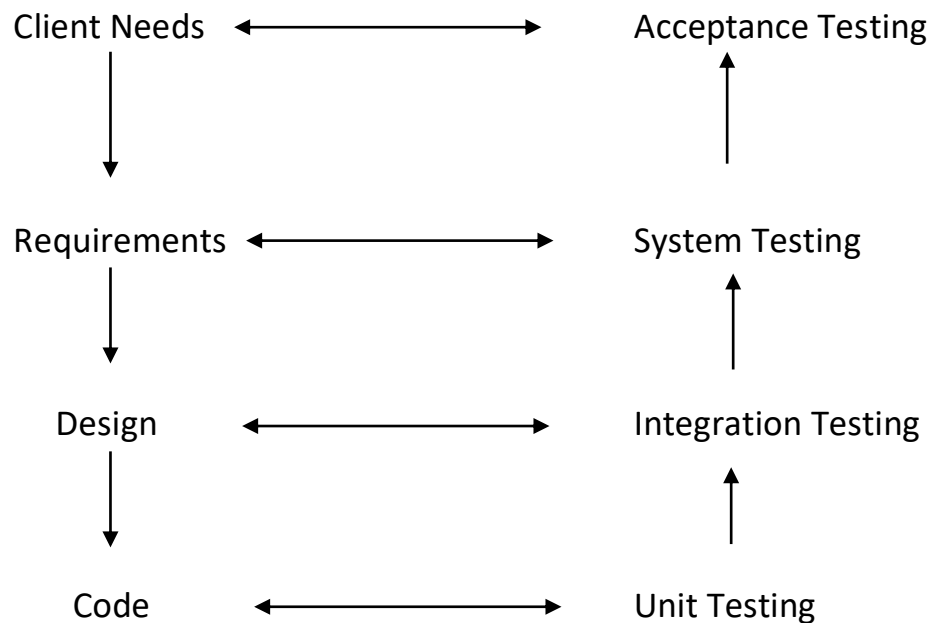
As the project is on a bit large scale, we always need testing to make it successful. If each component works properly in all respect and gives desired output for all kinds of inputs then the project is said to be successful. So, the conclusion is to make the project successful, it needs to be tested.

The testing done here was System Testing checking whether the user requirements were satisfied. The code for the new system has been written completely using Python as the coding language, and Django as the interface for front-end designing. The new system has been tested well with the help of the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms is very much by the actual flow of data.

Levels of Testing

To uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are:



A series of testing is done for the proposed system before the system is ready for user acceptance testing.

The steps involved in Testing are:

✓ **Unit Testing**

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as “Module Testing”. The modules are tested separately. This testing is carried out during the programming stage itself. In this testing, each module is found to be working satisfactorily as regards the expected output from the module.

✓ **Integration Testing**

Data can be grossed across an interface; one module can have adverse efforts on another. Integration testing is systematic testing for the construction of the program structure while at the same time conducting tests to uncover errors associated with the interface. The objective is to take unit-tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of cause is complicated by the vast expense of the entire program. Thus, in the integration testing stop, all the errors uncovered are corrected for the text testing steps.

✓ **System testing**

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then the goal will be successfully achieved.

✓ **Validation Testing**

After integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins, validation test begins. Validation tests can be defined in many ways. But the simple definition is that validation succeeds when the software function in a manner that can reasonably be expected by the customer. After the validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics confirm to specifications and are accepted and the other is a deviation from the specification is uncovered and a deficiency list is created. The proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

✓ **Output Testing**

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and the other is the printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user's needs.

For the hard copy also, the output comes as the specified requirements by the users. Hence output testing does not result in any corrections in the system.

✓ **User Acceptance Testing**

User acceptance of a system is the key factor to the success of any system. The system under study is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

Test Cases

Registration: To begin with login, users need to register by filling up basic registration details. There are multiple fields on the registration page and every field has to be filled by the user. The user cannot use characters in the login id field.

Login: - Login id and password are kept compulsory fields, and if the id or password doesn't match then it will show an error message.

VALIDATION CRITERIA

1. In each form, no field which is not nullable should be left blank.
2. All numeric fields should be checked for non-numeric values. Similarly, text fields like names should not contain any numeric characters.
3. All primary keys should be automatically generated to prevent the user from entering any existing key.
4. Use of error handling for each Save, Edit, delete and other important operations.
5. Whenever the user tabs out or enters from a text box, the data should be validated and if it is invalid, focus should again be sent to the text box with the proper message.

ADVANTAGES OF PROJECT

Advantages:

- The system provides round-the-clock assistance, allowing users to access healthcare information and support at any time, regardless of their location.
- It can provide users with accurate and up-to-date medical information.
- The system can also find doctors for users based on their location.
- It will also help users to book doctor appointments and further enhance accessibility through video calls.
- It works to enhance the efficiency of healthcare delivery, reduce waiting times, and improve overall patient satisfaction.

Limitations:

- Sometimes, the bot can struggle to fully comprehend the nuances and context of user queries.
- It can also struggle with rare or less common medical conditions that are not well-represented in their training data.

Features

1) Load Balancing:

Since the system will be available only after the admin logs in the amount of load on the server will be limited to the period of admin access.

2) Easy Accessibility:

Records can be easily accessed and stored and other information respectively.

3) User-Friendly:

The website/application will be giving a very user-friendly approach for all users.

4) Efficient and reliable:

Maintaining the secured database on the server which will be accessible according to the user requirement without any maintenance cost will be very efficient as compared to storing all the customer data on the spreadsheet or physically in the record books.

5) Easy maintenance:

AI Health Botis designed as an easy way. So, maintenance is also easy.

CONCLUSION

This was our project of System Design about “**AI Health Bot**” developed in Django in Python programming language. The Development of this system takes a lot of effort from us. We think this system gave a lot of satisfaction to all of us. Though every task is never said to be perfect in this development field even more improvement may be possible in this application. We learned so many things and gained a lot of knowledge about the development field. We hope this will prove fruitful to us.

BIBLIOGRAPHY

► Websites

- ✓ en.wikipedia.org
- ✓ <https://ieeexplore.ieee.org/document/9197833>
- ✓ <https://www.researchgate.net/publication/344983576> Chatbot for Healthcare System Using Artificial Intelligence
- ✓ <https://ijcrt.org/papers/IJCRT020011.pdf>
- ✓ <https://www.ijraset.com/research-paper/healthcare-chatbot-using-artificial-intelligence>
- ✓ <https://www.ijedr.org/papers/IJEDR2009083.pdf>